Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 9, No. 4, November, Pp. 353-366

Security analysis of web-based academic information system using OWASP framework

Rusydi Umar[1], Imam Riadi[2], Muhammad Ihya Aulia Elfatiha*[1]
Master Program of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia[1]
Department of Information System, Universitas Ahmad Dahlan, Yogyakarta, Indonesia[2]

Article Info

Keywords:
OWASP, Academic Information System, Penetration Testing, Security, Vulnerability

Article history:
Received: April 23, 2024
Accepted: June 25, 2024
Published: November 01, 2024

Cite:
Rusydi Umar, Imam Riadi, and M. I. A. Elfatiha, “Security Analysis of Web-based Academic Information System using OWASP Framework”, *KINETIK*, vol. 9, no. 4, Nov. 2024.
https://doi.org/10.22219/kinetik.v9i4.2015

*Corresponding author.
Muhammad Ihya Aulia Elfatiha
E-mail address:
elfatih2008048045@webmail.uad.ac.id

Abstract

The Academic Information System plays a crucial role in efficiently managing student, faculty, and campus administration data. However, system security needs to be a primary concern as it is vulnerable to cyber attacks. This research aims to analyze the security of the Academic Information System at the Muhammadiyah Business Institute Bekasi. The research method used is a comprehensive security analysis based on the OWASP framework. The study includes identifying potential vulnerabilities, penetration testing, and system improvement recommendations. Testing is conducted through simulated attacks based on the OWASP-released security risk list (OWASP Top Ten Most Critical Web Application Security Risks). The analysis results indicate that the system is vulnerable to Broken Authentication due to weak passwords, Sensitive Data Exposure due to URLs pointing to direct directories, and Security Misconfiguration due to open protocols. Furthermore, in CVSS scoring, Broken Authentication scored 4.8 (Medium), Sensitive Data Exposure and Security Misconfiguration scored 5.3 (Medium), Cross-Site Scripting scored 2.0 (Low) and Using Component with Known Vulnerabilities scored 2.0 (Low), while SQL Injection, XXE, Broken Access Control, Insecure Deserialization, and Insufficient Logging and Monitoring scored 0.0 (No Vulnerability). Recommendations for future system improvements include regularly updating the system to prevent new security vulnerabilities, better server configurations, and routine system monitoring to promptly anticipate suspicious activities.

1. Introduction

It cannot be denied that in today's age, every aspect of life has entered the era of digitalization, where various activities are conducted digitally [1]. The digital transformation in Indonesia is pushing all parties to make breakthroughs to avoid being left behind and experiencing decay [2]. Similarly, in the realm of higher education, the role of information technology in this era of digitalization is prompting universities to innovate in the face of intense competition to enhance the standards of higher education quality [3]. Anticipating the rapid dynamics of change and the increasingly complex challenges, universities need to strive for various means to enhance competitiveness and added value [4].

Institut Bisnis Muhammadiyah Bekasi (IBM Bekasi) has innovated by developing a system called AIS (Academic Information System). The system has been integrated with financial, personnel, e-learning modules, and feeder reporting modules, which are expected to serve as a role model for all universities, especially within the Muhammadiyah scope, due to the utilization of information systems that align with the vision and mission, as well as meeting the needs and reliability to provide significant benefits in achieving the goals of an organization [5], specifically within the Muhammadiyah higher education context.

However, issues related to information security are often overlooked, supported by the continued prevalence of information technologies that fail to provide beneficial effects for organizations, known as the "IT Productivity Paradox" [6][7]. IBM Bekasi has never conducted a comprehensive audit of the developed system, especially regarding its security aspects. Thus, up to this point, IBM Bekasi has been unable to determine the security of the implemented AIS. Therefore, based on the existing issues, this research analyzes the security of AIS owned by IBM Bekasi using penetration testing method with the OWASP framework.

After approximately two years of using AIS, there have been several complaints from users. These include grades not appearing on the student portal, grades inputted by lecturers not being saved, requiring operators to manually compile data for reporting, among others. Another common issue is the occurrence of login wait times, resulting in students being late for classes. These problems align with the statement by [8] that the more information stored and managed, the greater the risk of damage, loss, or unauthorized access to data by unwanted parties.

Based on the current situation, there are indicators indicating that the governance of the information system implemented at Institut Bisnis Muhammadiyah Bekasi is considered inadequate. Therefore, it is deemed necessary to have a mechanism such as an audit or analysis of Information Technology Governance (IT Governance) related to the current security system in order to improve it [9], thus obtaining a good information system in the future and in accordance with applicable standards [10].

In this study, the research will discuss the measurement of the security level of the IBM Bekasi AIS system using the OWASP framework [11], followed by Penetration Testing. The aim of the research is to obtain accurate and appropriate evaluations of the security system in the IBM Bekasi AIS [12]. Subsequently, conclusions will be drawn from the test results, identifying which parts of the system are vulnerable to attacks, serving as an evaluation reference for system improvement at IBM Bekasi in the future [10] to enhance system resilience.

2. Research Method

The process stages in this research refer to the OWASP (Open Web Application Security Project) framework, which focuses on application security [13]. OWASP produces various types of materials through collaboration and openness [14][15]. The use of OWASP as a framework can assist in identifying system vulnerabilities in detail. The following are the stages of OWASP framework as illustrated in Figure 1 [16].

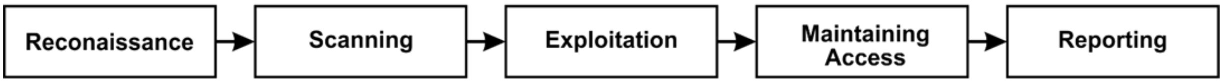
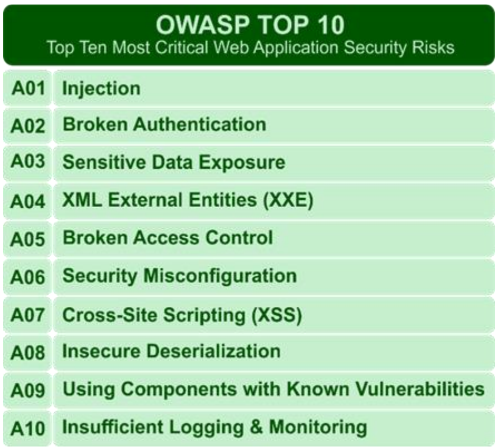


Figure 1. Stages of Open Web Application Security Project

The explanation of each stage in the OWASP framework as shown in Figure 1 consists of five main interconnected parts, namely:

1) Reconnaissance, which is the initial stage of data and information gathering related to the AIS system to be penetrated [17][18].
2) Scanning, which involves scanning to obtain deeper information and identify vulnerabilities in the AIS system [19][20].
3) Exploitation, which involves attacking the AIS system using data and information obtained during scanning [21].
4) Maintaining Access, which involves maintaining access gained by inserting backdoors through vulnerabilities in the system [22].
5) Reporting, which involves writing a report describing the testing results along with recommendations and solutions [23].

Referring to the OWASP Top 10, which is the Top Ten Most Critical Web Application Security Risks, there are ten lists of the highest security vulnerabilities released by OWASP most recently in 2017 as shown in Figure 2 [24][25].



OWASP TOP 10
Top Ten Most Critical Web Application Security Risks

A01	Injection
A02	Broken Authentication
A03	Sensitive Data Exposure
A04	XML External Entities (XXE)
A05	Broken Access Control
A06	Security Misconfiguration
A07	Cross-Site Scripting (XSS)
A08	Insecure Deserialization
A09	Using Components with Known Vulnerabilities
A10	Insufficient Logging & Monitoring

Figure 2. OWASP Top Ten Most Critical Web Application Security Risks

After the testing using OWASP Top 10 as shown in Figure 2, in addition to reporting the findings and providing advice or solutions to the findings in the system, the research also calculate how critical the security gap is by using the Common Vulnerability Scoring System (CVSS) Calculator [26] released by the First Organization as a vulnerability assessment system to measure the severity level of vulnerabilities in the system [27]. The assessment matrix available in CVSS can be seen in Figure 3.

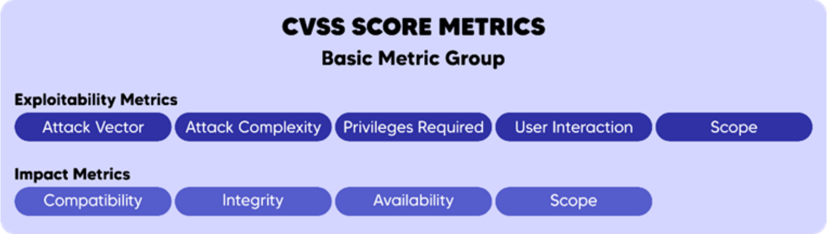


Figure 3. CVSS Score Metrics

In Figure 3, the assessment matrix is divided into two parts: exploitability metrics, which include attack vector, attack complexity, privileges required, and user interaction, and impact metrics, which include confidentiality impact, integrity impact, and availability impact. These attack and impact metrics are then adjusted according to the results obtained during the attack, and CVSS will calculate scores for each vulnerability identified. The assessment system can be seen in Table 1.

Table 1. Range Score CVSS.

Scoring	Score Range	Explanation
Low	0.1 – 3.9	Indicates that the system is still considered secure and no vulnerabilities are found.
Medium	4.0 – 6.9	Indicates that the system provides a responsive, albeit informational, feedback when exploited.
High	7.0 – 8.9	Indicates that the system responds by providing personal or sensitive data, even granting access to hackers when exploited.
Critical	9.0 – 10	Indicates that the system is highly vulnerable to malware infiltration and data theft, and can even be hijacked by hackers.

Based on Figure 1, Figure 2, Figure 3, and Table 1, this study designs a testing scenario for the academic information system as shown in Figure 4.

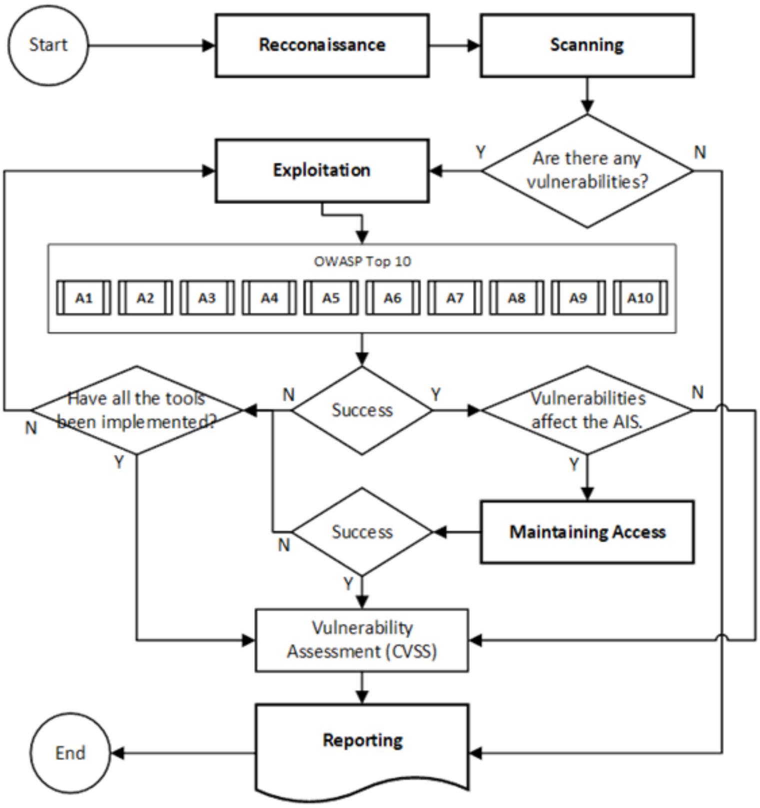

Start
Recconaissance
Scanning
Are there any vulnerabilities?
Y
N
Exploitation
OWASP Top 10
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
Have all the tools been implemented?
N
Y
Success
N
Y
Vulnerabilities affect the AIS.
N
Y
Maintaining Access
Success
N
Y
Vulnerability Assessment (CVSS)
Reporting
End

Figure 4. Attack Scenario

The sequence of attack scenarios shown in Figure 4 is described as follows:

1) In the reconnaissance phase, information gathering about the target is conducted using several tools, namely Netcraft to find the system's IP address, Whois to obtain domain information, Nmap to identify ports and services, and WhatWeb to identify the frameworks and plugins used.
2) In the scanning phase, vulnerabilities are identified using ZAP.
3) In the exploitation phase, after vulnerabilities are found, testing is conducted based on the OWASP Top 10 vulnerability list.
4) If a vulnerability is successfully exploited, the next step is maintaining access to further exploit and retain access to the system for a longer period.
5) After all testing based on the OWASP Top 10 vulnerability list is completed, the vulnerabilities are assessed using the CVSS Calculator to determine the rating of each vulnerability.
6) The final phase involves reporting all activities conducted, along with findings and recommendations for system improvements.

3. Results and Discussion

In this section, the discussion regarding the steps taken along with the research results on the subject is explained, followed by the compilation of findings into a report as well as recommendations and solutions based on the test results for future system improvements.

3.1 Recconaissance

This stage is often referred to as the Information Gathering stage, where researchers gather as much information as possible using non-technical methods such as search engines, email lists, and other public sources, as well as technical methods such as using tools available in Kali Linux.

3.1.1 Whois

Information gathering involves using the "whois" command via the Kali Linux terminal. This command is used to check information about the owner of a domain name or target IP address. Through the examination process carried out using the "whois" command, we can see detailed information such as hosting address, server address, email address, and other related information associated with the ibm.ac.id system. The results obtained from "whois" are shown in Table 2.

Table 2. The Results of the Whois Scanning Tool

Whois Domain Results		
Domain ID	:	PANDI-DO1241026
Domain Name	:	ibm.ac.id
Created On	:	2019-02-18 02:14:52
Last Updated On	:	2024-02-18 04:49:51
Expiration Date	:	2025-02-18 23:59:59
Name Server	:	dns1.masterweb.com dns2.masterweb.com dns3.masterweb.com dns4.masterwebnet.com
DNSSEC	:	Unsigned

3.1.2 Nmap

Network mapping is a tool used for network exploration and security auditing, where the results can be used as a basis for exploitation through available open ports. To use this tool in Kali Linux, the nmap command requires the target IP Address, in this case, the system has the IP Address 93.xxx.xxx.xx. Nmap will scan and provide information about the software versions and services connected to the IP address. The scanning results found on the server are further illustrated in Table 3.

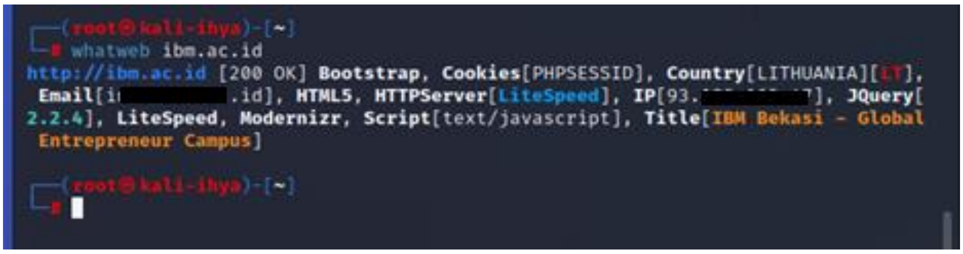
Table 3. The Results of the Nmap Scanning Tool

Port	State	Service	Version
22/tcp	open	ssh	OpenSSH 8.9p1 3ubuntu0.5 (Ubuntu Linux; protocol 2.0)
53/tcp	open	domain	PowerDNS Authoritative Server 4.5.3
80/tcp	open	http	LiteSpeed
443/tcp	open	tcpwrapped	

In addition to the results explained in Table 3, further scanning of the system was conducted using the command -v -A to obtain information about the OS and other details. The scan successfully displayed service information, indicating that the host used is srv427307, the server is LiteSpeed, the Signature Algorithm is present, the operating system is Linux, and the Terminal/CPE is cpe:/o:linux:linux_kernel.

3.1.3 WhatWeb

The WhatWeb scan is used to determine the type of framework used by analyzing the HTTP headers, cookies, source code, and specific files and folders on ibm.ac.id. The scan was performed on Kali Linux as shown in Figure 5.



```
┌──(root㉿kali-ihya)-[~]
└─# whatweb ibm.ac.id
http://ibm.ac.id [200 OK] Bootstrap, Cookies[PHPSESSID], Country[LITHUANIA][LT],
 Email[i[redacted].id], HTML5, HTTPServer[LiteSpeed], IP[93.[redacted]], JQuery[
2.2.4], LiteSpeed, Modernizr, Script[text/javascript], Title[IBM Bekasi - Global
 Entrepreneur Campus]

┌──(root㉿kali-ihya)-[~]
└─# █
```

Figure 5. The Results of the Whatweb Tool

The result obtained in Figure 5 is a summary of the website, with comprehensive information gathered using the -v -a command in whatweb. It indicates that the website uses Bootstrap [5.0.0] as the framework, Email [info@ibm.ac.id] as the email address used, HTML5 as the markup language utilized, HTTPServer [LiteSpeed] as the server, and Jquery [2.2.4] as the JavaScript library employed.

The conclusion of the information gathering is presented in Table 4.

Table 4. System Information Gathering Results

No	Process	Result
1	Netcraft	IP Address: 93.xxx.xxx.xx
2	Whois	Hosting address, server, e-mail, and other related information regarding ibm.ac.id.
3	Nmap	Open port: Port 22/SSH, Port 53/Domain, Port 80/http
4	Whatweb	HTTP Header

3.2 Scanning

Vulnerability scanning is conducted to discover security vulnerabilities on system using the Zaproxy tool on Kali Linux based on the risk level of each vulnerability finding. The scanning results are shown in Table 5.

Table 5. The Results of the Zaproxy Scanning

No	Alerts	Total	Risk
1	Directory Browsing	8	Medium
2	CSP Header Not Set	4	Medium
3	Hidden File Found	1	Medium
4	Cross-Domain Misconfiguration	1	Medium
5	Missing Anti-clickjacking Header	3	Medium
6	Anti-CSRF Tokens Check	3	Medium
7	Vulnerable JS Library	3	Medium
8	Cookie No HttpOnly Flag	2	Low
9	Cookie without SameSite Attribute	2	Low
10	X-Content-Type-Options Header Missing	16	Low
11	Authentication Request Identified	1	Informational
12	Informational Disclosure – Suspicious Comment	3	Informational
13	Modern Web Application	3	Informational
14	Retrieved from Cache	3	Informational
15	Session Management Response Identified	19	Informational

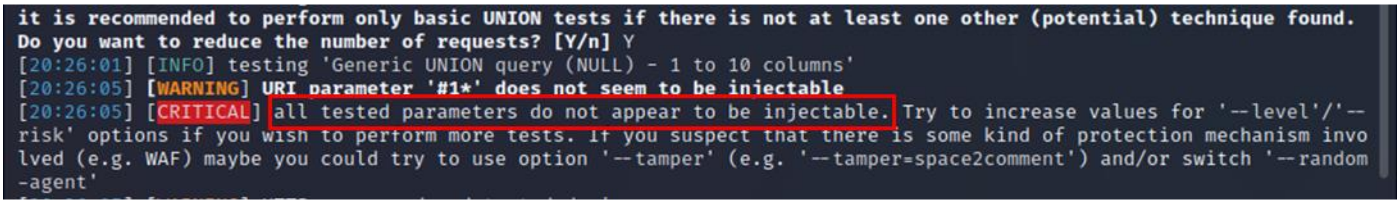
In the scanning results presented in Table 5, no high/critical-level vulnerabilities were found. This indicates that the system is relatively safe from attacks during the initial scanning and attack stages. However, in the next stage to assess the system's security further, exploitation will be conducted using penetration testing methods by injecting SQL syntax (SQL Injection) and Cross-Site Scripting (XSS). This is because these types of attacks are commonly attempted by hackers and are vulnerable to exploitation.

3.3 Exploitation

In the exploitation phase, the targeted system will be infiltrated by simulating attacks to identify security vulnerabilities. In this phase, evaluation is conducted using the OWASP Top 10 Most Critical Web Application Security Risks list to discover weaknesses in the system's security. The goal is to test the system's security and identify vulnerabilities that can be exploited by researchers.

3.3.1 SQL Injection

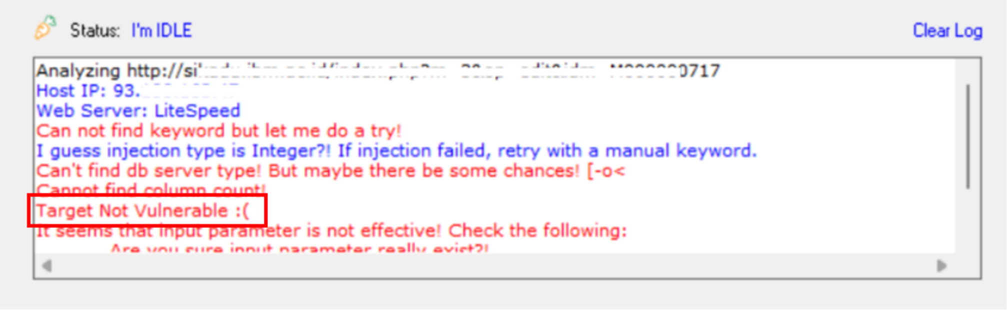
To conduct a SQL Injection Attack, the tool used is SQLMap, which can be found in the Kali Linux operating system. The goal of this attack is to retrieve all the information stored in the database system. To execute the attack, the command used is -u http://xxxxxxx.ibm.ac.id/latest/meta-data --dbs. as seen in Figure 6.



```
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found.
Do you want to reduce the number of requests? [Y/n] Y
[20:26:01] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[20:26:05] [WARNING] URI parameter '#1*' does not seem to be injectable
[20:26:05] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--
risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism invo
lved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random
-agent'
```

Figure 6. SQL Injection Using SQLMap

From the test results in Figure 6, there is a statement "all tested parameters do not appear to be injectable," which can be concluded that no vulnerabilities were found for injection attacks on http://xxxxxx.ibm.ac.id due to being blocked by the web application firewall (WAF) owned by the target website. Then, the researcher tried to conduct further exploration using Havij as shown in Figure 7.



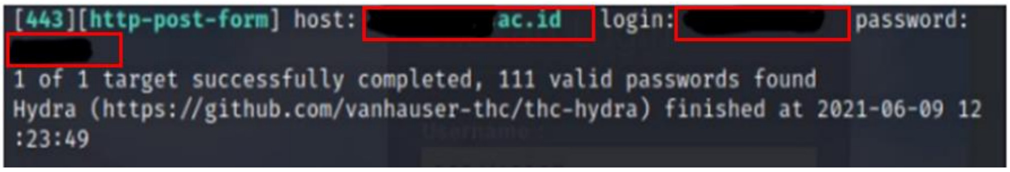
```
Analyzing http://si[redacted]0717
Host IP: 93.[redacted]
Web Server: LiteSpeed
Can not find keyword but let me do a try!
I guess injection type is Integer?! If injection failed, retry with a manual keyword.
Can't find db server type! But maybe there be some chances! [-o<
Cannot find column count!
Target Not Vulnerable :(
It seems that input parameter is not effective! Check the following:
    Are you sure input parameter really exist?!
```

Figure 7. Injection with Havij Tool

The injection process conducted in Figure 7 resulted in Havij tool not being able to find the database server on the ID parameter obtained from the Zaproxy scanning results, with the analysis indicating that the Target is Not Vulnerable. Therefore, it can be concluded that the system is safe from SQL injection attacks.

3.3.2 Broken Authentication

In this test, to exploit the Broken Authentication vulnerability on the target system, the Hydra Tool available in Kali Linux is used. Hydra Tool is used to perform brute force tailored to the target specifications, in this case, the system's login page. The results of the brute force attack can be seen in Figure 8.



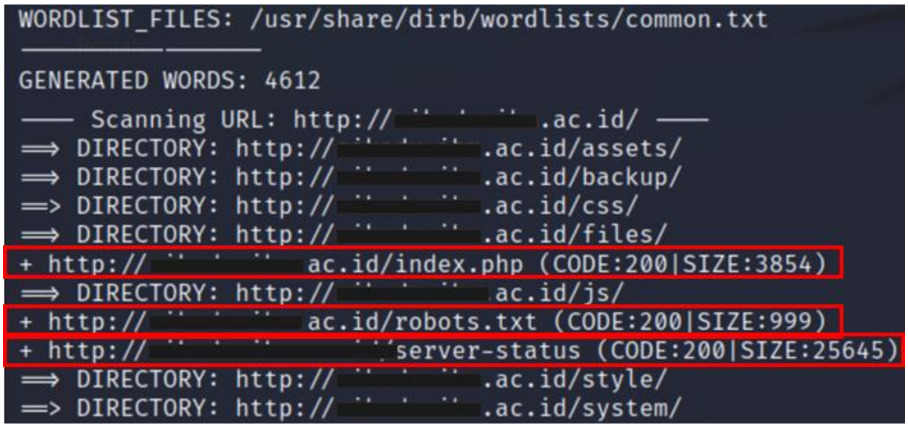
```
[443][http-post-form] host: [redacted]ac.id   login: [redacted]   password:
[redacted]
1 of 1 target successfully completed, 111 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-06-09 12
:23:49
```

Figure 8. Broken Authetication Attack with Hydra

From the results of the attack attempt in Figure 8, it can be concluded that the brute-force attack was successful, and a vulnerability was found in sikad.ibm.ac.id. It was found that several usernames and passwords were discovered, along with some HTTP post information from the scanning results using Burp Proxy. Some of these usernames and passwords can be used to attempt to log in to the system.

3.3.3 Sensitive Data Exposure

This stage tests the security vulnerabilities in the system that could lead to the leakage of sensitive information such as personal data and passwords. The testing is conducted using Dirb by entering the command dirb http://xxxxxx.ibm.ac.id as shown in Figure 9.



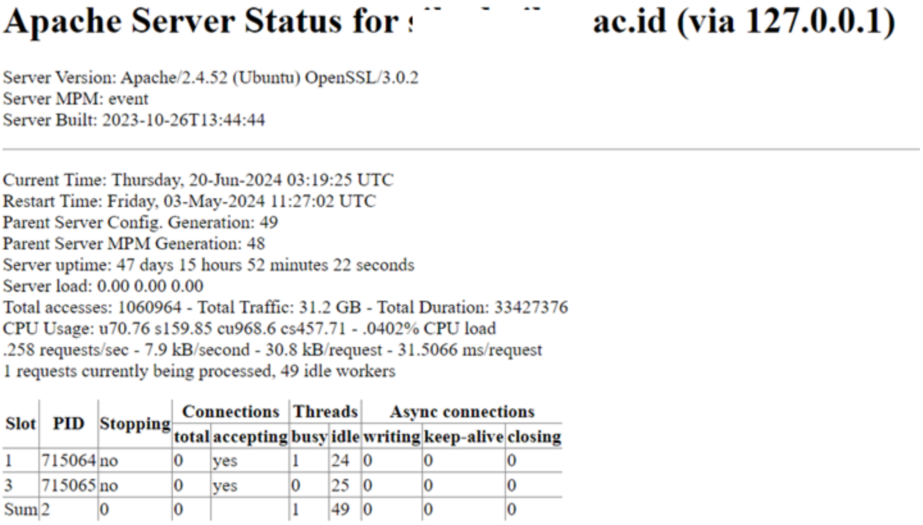
```
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
─────────────────

GENERATED WORDS: 4612

──── Scanning URL: http://[redacted].ac.id/ ────
==> DIRECTORY: http://[redacted].ac.id/assets/
==> DIRECTORY: http://[redacted].ac.id/backup/
==> DIRECTORY: http://[redacted].ac.id/css/
==> DIRECTORY: http://[redacted].ac.id/files/
+ http://[redacted]ac.id/index.php (CODE:200|SIZE:3854)
==> DIRECTORY: http://[redacted]ac.id/js/
+ http://[redacted]ac.id/robots.txt (CODE:200|SIZE:999)
+ http://[redacted]'server-status (CODE:200|SIZE:25645)
==> DIRECTORY: http://[redacted].ac.id/style/
==> DIRECTORY: http://[redacted].ac.id/system/
```

Figure 9. Results of Sensitive Data Exposure Identification using the Dirb Tool

The test results in Figure 9 show that files in the system directory can be directly accessed with status code 200 for index.php, robot.txt, and server-status. The URL index.php is a commonly used login page. The URL robot.txt does not respond, while the URL server-status displays server information on port 80. The test results for the URL http://xxxxxx.ibm.ac.id/serverstatus are shown in Figure 10.



Apache Server Status for ac.id (via 127.0.0.1)

Server Version: Apache/2.4.52 (Ubuntu) OpenSSL/3.0.2
Server MPM: event
Server Built: 2023-10-26T13:44:44

Current Time: Thursday, 20-Jun-2024 03:19:25 UTC
Restart Time: Friday, 03-May-2024 11:27:02 UTC
Parent Server Config. Generation: 49
Parent Server MPM Generation: 48
Server uptime: 47 days 15 hours 52 minutes 22 seconds
Server load: 0.00 0.00 0.00
Total accesses: 1060964 - Total Traffic: 31.2 GB - Total Duration: 33427376
CPU Usage: u70.76 s159.85 cu968.6 cs457.71 - .0402% CPU load
.258 requests/sec - 7.9 kB/second - 30.8 kB/request - 31.5066 ms/request
1 requests currently being processed, 49 idle workers

Slot	PID	Stopping	Connections		Threads		Async connections		
			total	accepting	busy	idle	writing	keep-alive	closing
1	715064	no	0	yes	1	24	0	0	0
3	715065	no	0	yes	0	25	0	0	0
Sum	2	0	0		1	49	0	0	0

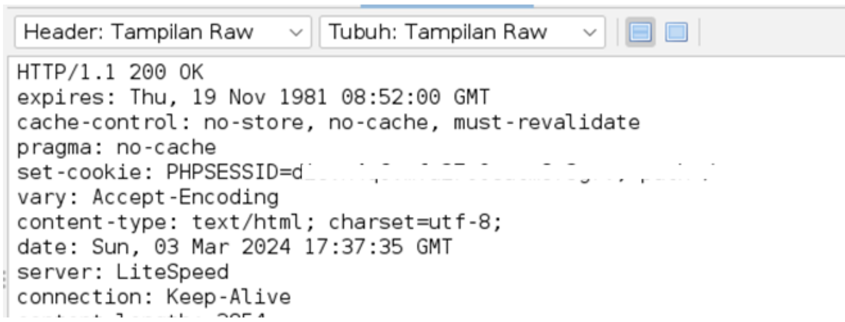
Figure 10. Test Results of the URL Indicating Sensitive Data Exposure

The test results in Figure 9 indicate that the URL leading to server-status can be accessed directly by the public without authentication. This URL is potentially vulnerable because it displays detailed information about the Apache server status. Essentially, Apache's server status feature is provided to give detailed information about the server performance and operational status for system administrators' monitoring and troubleshooting purposes, but it can become problematic if not configured correctly.

As found in this research, the server status can be accessed by anyone who knows the URL, exposing detailed information such as total accesses and traffic, CPU usage, uptime, requests per second, worker status, and the scoreboard. With these findings, it can be concluded that the system has a vulnerability in Sensitive Data Exposure because it does not restrict access to sensitive data. The URL should be reconfigured to ensure that only specific IP addresses are allowed to access that page. For system security, the file should be configured so that it cannot be accessed by the public or attackers.

3.3.4 XML External Entities (XXE)

XXE (XML External Entity) testing is conducted to verify whether the system responds to or rejects injected XXE documents, and whether there is validation and filtering on the target website or not, by reviewing the Zaproxy scanning results as shown in Figure 11.



```
HTTP/1.1 200 OK
expires: Thu, 19 Nov 1981 08:52:00 GMT
cache-control: no-store, no-cache, must-revalidate
pragma: no-cache
set-cookie: PHPSESSID=d[redacted]
vary: Accept-Encoding
content-type: text/html; charset=utf-8;
date: Sun, 03 Mar 2024 17:37:35 GMT
server: LiteSpeed
connection: Keep-Alive
content-length: 2054
```

Figure 11. XML File Scanning Result

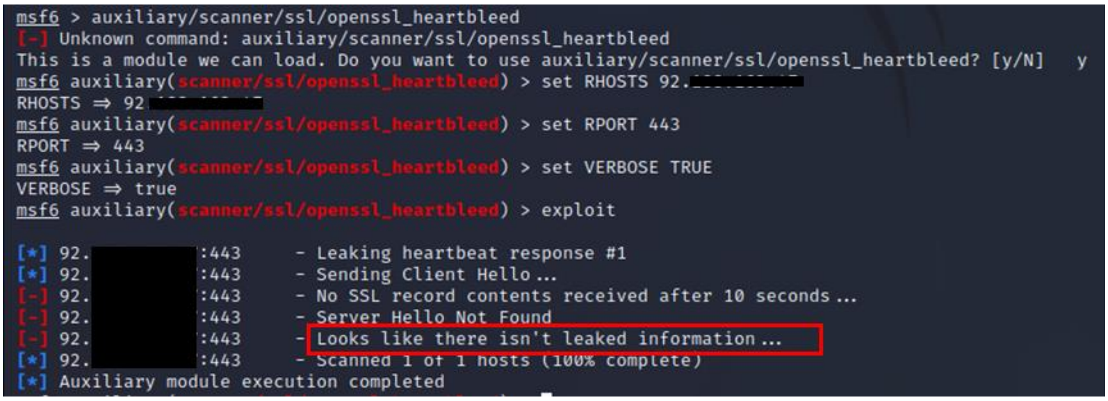
In Figure 11, it can be seen that the request results indicate no XML tags were found. To execute an XXE attack, the presence of XML tags is required, and if not found, they can be inserted. However, attempts made several times to modify the XML tags yielded no response from system, thus it can be concluded that the XXE attack was not successful on the targeted system

3.3.5 Broken Access Control

Broken access control or missing function level access control is the illegal process of altering access rights without the system administrator's knowledge by exploiting vulnerable URLs obtained during ZAP scanning phase. However, in the ZAP scan results, there are no URLs in the system displaying the ID parameter, hence manual access attempts cannot be performed. Therefore, it can be concluded from the ZAP results that there is no access available to carry out attacks on broken access control.

3.3.6 Security Missconfiguration

Security misconfiguration is a vulnerability that occurs when a system is not properly configured, leaving security gaps that can be exploited by attackers. Table 5 of the ZAP scan results did not detect any Security Misconfiguration vulnerabilities. Therefore, this stage requires further testing using the Metasploit tool, leveraging the Heartbleed Library Bug feature. The results of this testing are shown in Figure 12.



```
msf6 > auxiliary/scanner/ssl/openssl_heartbleed
[-] Unknown command: auxiliary/scanner/ssl/openssl_heartbleed
This is a module we can load. Do you want to use auxiliary/scanner/ssl/openssl_heartbleed? [y/N]   y
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set RHOSTS 92.[redacted]
RHOSTS ⇒ 92[redacted]
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set RPORT 443
RPORT ⇒ 443
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > set VERBOSE TRUE
VERBOSE ⇒ true
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > exploit

[*] 92.[redacted]:443     - Leaking heartbeat response #1
[*] 92.          :443     - Sending Client Hello ...
[-] 92.          :443     - No SSL record contents received after 10 seconds ...
[-] 92.          :443     - Server Hello Not Found
[-] 92.          :443     - Looks like there isn't leaked information ...
[*] 92.          :443     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 12. Results of Security Misconfiguration Testing using Metasploit Heartbleed Library Bug

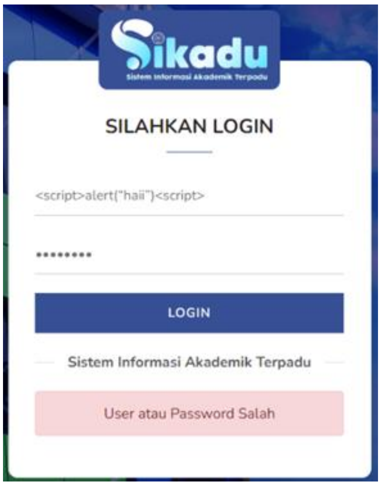
The result of exploiting the Heartbleed bug shown in Figure 12 states that “looks like there isn’t leaked information,” which means there is no leaked confidential information that could pose a threat through the SSL/HTTP port. It can be concluded that there are no configuration errors on the server

3.3.7 Cross-Site Scripting (XSS)

XSS attacks only require the insertion of scripts or HTML on a web page. Typically, when testing for XSS attack possibilities, input validation authenticity needs to be verified, and testers should be aware of this when validating the output from the website. Additionally, when reviewing code, it's important to understand how input matches to output.

a. Manual Input Script

The syntax that can be tried on sikad.ibm.ac.id includes <script>alert(“hello”)<script>. The attempt is made on the web login page as shown in Figure 13.



SILAHKAN LOGIN

<script>alert("haii")<script>

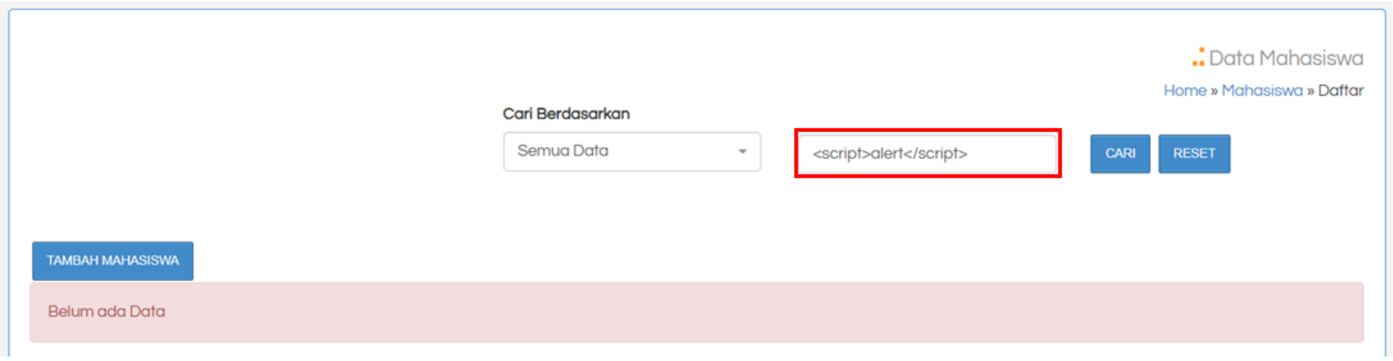
••••••••

Sistem Informasi Akademik Terpadu

User atau Password Salah

Figure 13. Testing XSS on the Login Page

Figure 13 shows an attempt of XSS Attack on login form. It can be observed that the sent script received no response and did not display anything on the login page. Subsequently, the attempt continued on the search column within the system, and the results of the attempt are shown in Figure 14.



Data Mahasiswa

Home » Mahasiswa » Daftar

Cari Berdasarkan

Semua Data

<script>alert</script>

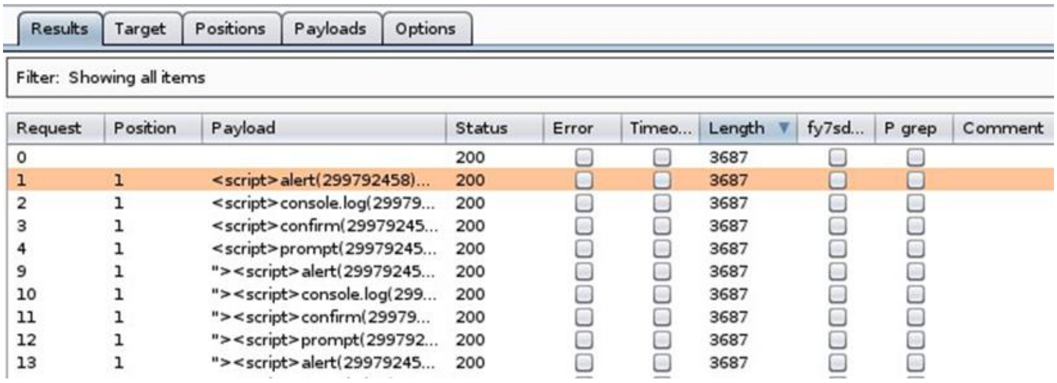
Belum ada Data

Figure 14. Testing XSS on the Search Field

Figure 14 is the result of the XSS syntax attempt conducted on the search column of the sikad.ibm.ac.id application. It can be seen that no output appears, indicating that from the XSS attempt on the system, there is no vulnerability to perform XXS, and it can be said that the sikad.ibm.ac.id website is safe from XSS attacks.

b. Input Script with Burpsuite

Another XSS attack attempt can be conducted by leveraging scripts and keys obtained from Burpsuite. This is done because the scripts and keys available in Burpsuite are more diverse, thus increasing the likelihood of discovering vulnerabilities on the targeted website. The test results can be seen in Figure 15.



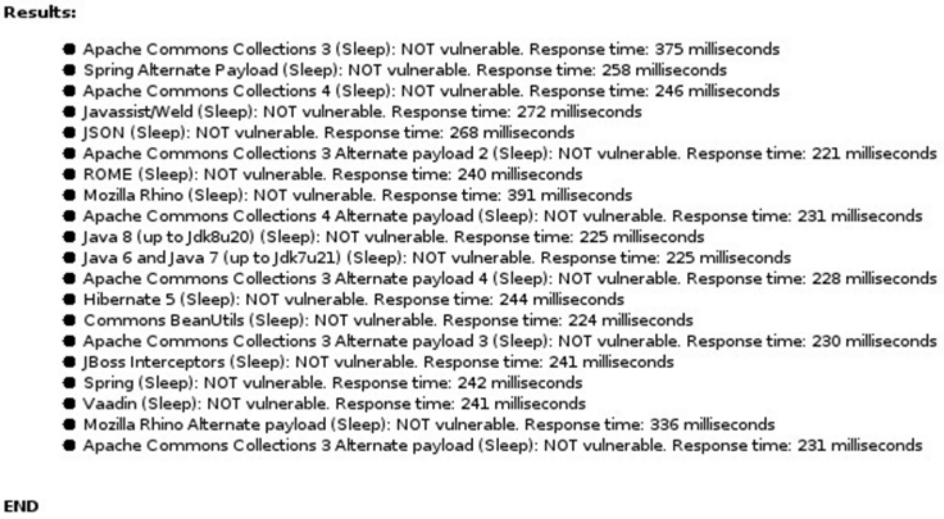
Request	Position	Payload	Status	Error	Timeo...	Length	fy7sd...	P grep	Comment
0			200	□	□	3687	□	□	
1	1	<script>alert(299792458)...	200	□	□	3687	□	□	
2	1	<script>console.log(29979...	200	□	□	3687	□	□	
3	1	<script>confirm(29979245...	200	□	□	3687	□	□	
4	1	<script>prompt(29979245...	200	□	□	3687	□	□	
9	1	"><script>alert(29979245...	200	□	□	3687	□	□	
10	1	"><script>console.log(299...	200	□	□	3687	□	□	
11	1	"><script>confirm(29979...	200	□	□	3687	□	□	
12	1	"><script>prompt(299792...	200	□	□	3687	□	□	
13	1	"><script>alert(29979245...	200	□	□	3687	□	□	

Figure 15. Testing XSS with Burpsuite Library

From Figure 15, it can be observed that although the script request was successfully sent to the system, there was no response displaying feedback on the XSS attack attempt. This indicates that despite using various syntaxes and keys available in Burpsuite, the XSS attack was still failed or unsuccessful. This means that the system is secure from any form of XSS attack.

3.3.8 Insecure Deserialization

In this stage, attempts to exploit insecure deserialization vulnerabilities were made using the Burp Suite tool extended with java deserialization.jar, as shown in Figure 16.



Results:

- Apache Commons Collections 3 (Sleep): NOT vulnerable. Response time: 375 milliseconds
- Spring Alternate Payload (Sleep): NOT vulnerable. Response time: 258 milliseconds
- Apache Commons Collections 4 (Sleep): NOT vulnerable. Response time: 246 milliseconds
- Javassist/Weld (Sleep): NOT vulnerable. Response time: 272 milliseconds
- JSON (Sleep): NOT vulnerable. Response time: 268 milliseconds
- Apache Commons Collections 3 Alternate payload 2 (Sleep): NOT vulnerable. Response time: 221 milliseconds
- ROME (Sleep): NOT vulnerable. Response time: 240 milliseconds
- Mozilla Rhino (Sleep): NOT vulnerable. Response time: 391 milliseconds
- Apache Commons Collections 4 Alternate payload (Sleep): NOT vulnerable. Response time: 231 milliseconds
- Java 8 (up to Jdk8u20) (Sleep): NOT vulnerable. Response time: 225 milliseconds
- Java 6 and Java 7 (up to Jdk7u21) (Sleep): NOT vulnerable. Response time: 225 milliseconds
- Apache Commons Collections 3 Alternate payload 4 (Sleep): NOT vulnerable. Response time: 228 milliseconds
- Hibernate 5 (Sleep): NOT vulnerable. Response time: 244 milliseconds
- Commons BeanUtils (Sleep): NOT vulnerable. Response time: 224 milliseconds
- Apache Commons Collections 3 Alternate payload 3 (Sleep): NOT vulnerable. Response time: 230 milliseconds
- JBoss Interceptors (Sleep): NOT vulnerable. Response time: 241 milliseconds
- Spring (Sleep): NOT vulnerable. Response time: 242 milliseconds
- Vaadin (Sleep): NOT vulnerable. Response time: 241 milliseconds
- Mozilla Rhino Alternate payload (Sleep): NOT vulnerable. Response time: 336 milliseconds
- Apache Commons Collections 3 Alternate payload (Sleep): NOT vulnerable. Response time: 231 milliseconds

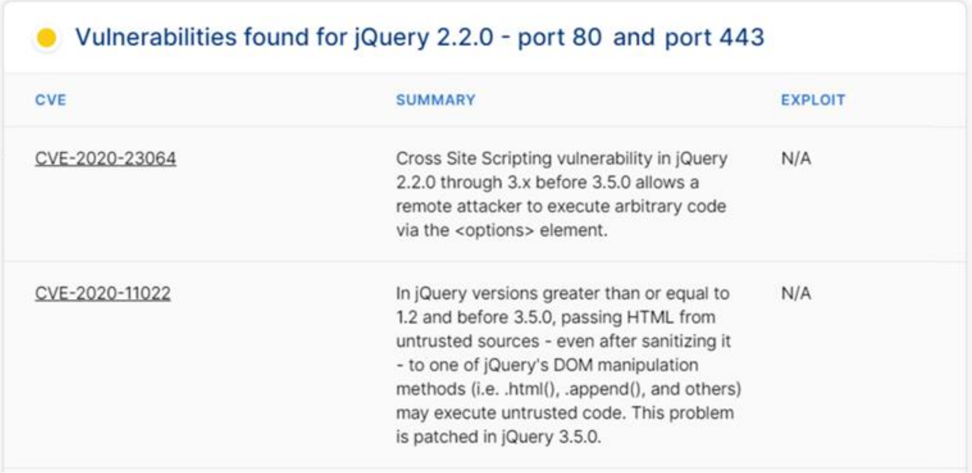
END

Figure 16. Insecure Deserialization Test using Burpsuite Tool

The test results in Figure 16 show that there are no vulnerabilities for conducting insecure deserialization attacks. This means the system is safe from the threat of deserialization attacks.

3.3.9 Using Components with Known Vulnerabilities

Using Components with Known Vulnerabilities is a vulnerability that occurs when a system uses libraries, frameworks, or other components that are outdated or have known vulnerabilities. In the ZAP scanning results in Table 5, there is a Vulnerable JS Library vulnerability identified. The evidence of the vulnerable JavaScript library is shown in Figure 17.



● Vulnerabilities found for jQuery 2.2.0 - port 80 and port 443

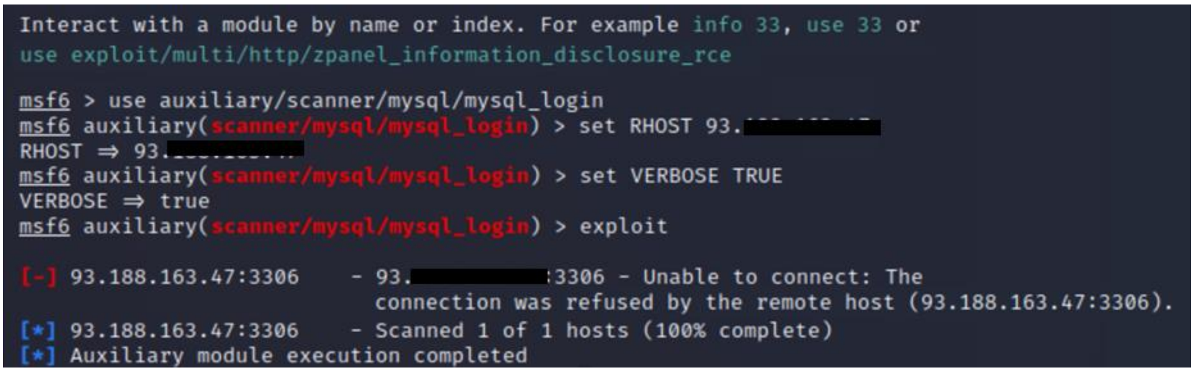
CVE	SUMMARY	EXPLOIT
CVE-2020-23064	Cross Site Scripting vulnerability in jQuery 2.2.0 through 3.x before 3.5.0 allows a remote attacker to execute arbitrary code via the <options> element.	N/A
CVE-2020-11022	In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.	N/A

Figure 17. Vulnerable JS Library

Based on the testing in Figure 17, it was found that the system uses outdated components, specifically jQuery version 2.2.0, whereas the latest version is jQuery 3.7.1. This outdated JavaScript library has vulnerabilities that can be exploited by attackers, such as Denial of Service (DoS) attacks, Cross-Site Scripting (XSS), and others.

3.3.10 Insufficient Logging and Monitoring

The maintaining access stage is where the research leverages the vulnerabilities found further. In this test, the Metasploit tool is used. The results of maintaining access on the identified vulnerability using Metasploit can be seen in Figure 18.



```
Interact with a module by name or index. For example info 33, use 33 or
use exploit/multi/http/zpanel_information_disclosure_rce

msf6 > use auxiliary/scanner/mysql/mysql_login
msf6 auxiliary(scanner/mysql/mysql_login) > set RHOST 93.[redacted]
RHOST ⇒ 93.[redacted]
msf6 auxiliary(scanner/mysql/mysql_login) > set VERBOSE TRUE
VERBOSE ⇒ true
msf6 auxiliary(scanner/mysql/mysql_login) > exploit

[-] 93.188.163.47:3306    - 93.[redacted]:3306 - Unable to connect: The
                            connection was refused by the remote host (93.188.163.47:3306).
[*] 93.188.163.47:3306    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 18. Logging and Monitoring Attack

The testing in Figure 18 shows a result indicating "Unable to connect: The connection was refused by the remote host (93.xxx.xxx.xx:3306)", indicating a rejection of the sent payload. The web server intercepted the payload sent to the system, resulting in immediate denial/blockage of access to the system. With this result, it can be concluded that the system has reasonably good security as the web server can detect suspicious activities on the system and promptly reject unauthorized access requests.

3.4 Reporting

The final stage is to create a report regarding the test results. The test results are outlined based on the OWASP Top 10 (Most Critical Web Application Security Risks). The report that can be presented can be seen in Table 6.

Table 6. Testing Report based on OWASP Top 10

Vulnerability	Status	Results	Solutions
A01: Injection	Not Found	All tested parameters do not appear to be injectable, Target Not Vulnerable.	
A02: Broken Authentication	Found	Hydra: “1 of 1 target successfully completed, 111 valid passwords found”	Use strong passwords, and implement the "secure" option in cookie settings.

A03: Sensitive Data Exposure	Found	Directory: http://xxxxxx.ibm.ac.id/server-status	The URL needs to be reconfigured to ensure that only specific IP addresses are allowed to access that page. For system security, the files should be configured so that neither the public nor attackers can access those sensitive files.
A04: XML External Entities	Not Found	No XML files were found, and attempts to inject XML files into the system were unsuccessful	
A05: Broken Access Control	Not Found	No suitable ID parameter found to conduct the attack.	
A06: Security Misconfiguration	Not Found	Looks like there isn't leaked information	
A07: Cross-Site Scripting (XSS)	Not Found	There was no response displaying feedback for the XSS attack attempt.	
A08: Insecure Deserialization	Not Found	Not Vulnerable to Attacks.	
A09: Using Component with Known Vulnerabilities	Found	The system has an outdated JavaScript library, where the JavaScript used by the system is jQuery version 2.2.0.	Update JS Library regularly and use stable and tested versions (avoid using Beta versions). For checking, you can use OWASP Dependency-Check or Retire.js.
A10: Insufficient Logging and Monitoring	Not Found	Unable to connect: The connection was refused by the remote host (93.xxx.xxx.xx:3306)	

Then, based on the obtained results, to assess the level of risk and the impact of the arising attacks, calculations were made using a special calculator from NIST (National Institute of Standards and Technology) called CVSS (Common Vulnerability Scoring System) with a score range from 0.0 to 10.0. The results obtained from the CVSS calculation can be seen in Table 7.

Table 7. CVSS Score

Vulnerability by OWASP Top 10	CVSS Score	Risk Rating
A1: SQL Injection	0.0	None
A2: Broken Authentication	4.8	Medium
A3: Sensitive Data Exposure	5.3	Medium
A4: XML External Entities (XXE)	0.0	None
A5: Broken Access Control	0.0	None
A6: Security Misconfiguration	5.3	Medium
A7: Cross-site Scripting (XSS)	2.0	Low
A8: Insecure Deserialization	0.0	None
A9: Using Component with Known Vulnerabilities	2.0	Low
A10: Insufficient Logging and Monitoring	0.0	None

From Table 7, it can be observed that the academic information system owned by the Institute Business Muhammadiyah Bekasi has a relatively low level of risk, indicating that the system is still considered safe.

4. Conclusion

From the conducted research, the penetration testing process using the parameters of the OWASP Top 10 Most Critical Web Application Security Risks revealed several findings indicating that the system has security vulnerabilities but still falls within the safe category. From the testing conducted, the results showed that for injection, the system used query parameters and input validation as measures to prevent SQL Injection attacks. For broken authentication, the system employed session management despite instances of weak password usage. In terms of sensitive data exposure, data encryption and user access levels were implemented, ensuring that only authorized users could control certain data. The XXE testing indicated that the system successfully rejected XML file insertion attempts. Broken access control was mitigated by implementing strong access controls. However, security misconfigurations were still present, with some configurations remaining open (enabled). For XSS testing, the system applied a Web Application Firewall (WAF), preventing any response to input attempts. Insecure deserialization showed no vulnerabilities in the system. However, Using Components with Known Vulnerabilities revealed that the system did not use the latest jQuery version, making it susceptible to attacks. Lastly, insufficient logging and monitoring were noted as the attempted attacks did not succeed. For future system improvements, this study recommends several points to consider, including periodic system updates to prevent new security vulnerabilities, closing open ports, and regular system monitoring to promptly address any suspicious activities.

The OWASP framework is highly effective in analyzing the security of web-based academic information systems. This framework provides a structured methodology for identifying, classifying, and prioritizing potential security risks. OWASP regularly releases a list of web application security risks, known as the OWASP Top 10, to keep pace with technological advancements. Additionally, the OWASP framework offers comprehensive security testing tools and guidelines for web applications, along with mitigation strategies for identified security risks. By implementing OWASP as a framework for conducting web application security audits, developers can enhance system resilience against cyber attacks and ensure better data protection.

References

[1] Mujiyono, “Jurnal Ilmiah Pendidikan Citra Bakti || 75,” *J. Ilm. Pendidik. citra Bakti*, vol. 6, no. 1, pp. 75–86, 2019.

[2] H. Muchsin, “Peluang dan Tantangan Perguruan Tinggi Menghadapi Revolusi Digital di Era Society 5.0,” *Pros. Semin. Nas. Pendidik.*, pp. 350–355, 2021.

[3] B. Rajagukguk, “Paradigma baru dalam meningkatkan mutu pendidikan,” *J. Tabularasa*, vol. 6, no. 1, pp. 77–86, 2009.

[4] M. Suti, M. Z. Syahdi, and D. D., “Tata Kelola Perguruan Tinggi dalam Era Teknologi Informasi dan Digitalisasi,” *JEMMA (Journal Econ. Manag. Accounting)*, vol. 3, no. 2, p. 203, 2020. https://doi.org/10.35914/jemma.v3i2.635

[5] M. Ratnasari, “Perencanaan Strategis Sistem Informasi/Teknologi Informasi Dalam Menghadapi Persaingan Pendidikan Tinggi Ilmu Komputer,” *J. Darma Agung*, vol. 30, no. 1, pp. 949–958, 2022.

[6] M. H. Murdani, M. U. Sari, and M. Muharom, “IT Productivity Paradox pada Perguruan Tinggi Swasta,” *J. Ilm. Teknol. Inf. Asia*, vol. 12, no. 2, p. 81, 2018. https://doi.org/10.32815/jitika.v12i2.216

[7] R. gilang jodi Putra, *Paradoks Produktivitas Teknologi Investasi Sistem Aplikasi Crm (Studi Kasus : Productivity Paradox of Information Technology : Investment Analysis of Crm Application System (Case Study : Pt . Xyz)*. 2015.

[8] Y. Darmawan and F. A. Wijaya, “Analisis Sistem Manajemen Keamanan Informasi Pada Perguruan Tinggi Menggunakan Iso 27001 : 2013,” *Semin. Nas. Sist. Inf. Indones.*, no. November, pp. 6–7, 2017.

[9] F. P. Utama, R. Muhamad, and H. Nurhadi, “Uncovering the Risk of Academic Information System Vulnerability through PTES and OWASP Method,” *CommIT J.*, vol. 18, no. 1, pp. 39–51, 2024. https://doi.org/10.21512/commit.v18i1.9384

[10] H. F. Tipton and M. Krause, *Information security management handbook*, vol. 2. 2008. https://doi.org/10.1201/9781420067101

[11] I. Riadi, A. Fadlil, and M. A. Mu’min, “OWASP Framework-based Network Forensics to Analyze the SQLi Attacks on Web Servers,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 22, no. 3, pp. 481–494, 2023. https://doi.org/10.30812/matrik.v22i3.3018

[12] A. Fadlil, I. Riadi, and M. A. Mu’Min, “Mitigation from SQL Injection Attacks on Web Server using Open Web Application Security Project Framework,” *Int. J. Eng. Trans. A Basics*, vol. 37, no. 4, pp. 635–645, 2024. https://doi.org/10.5829/ije.2024.37.04a.06

[13] I. F. Ashari, V. Oktarina, R. G. Sadewo, and S. Damanhuri, “Analysis of Cross Site Request Forgery (CSRF) Attacks on West Lampung Regency Websites Using OWASP ZAP Tools,” *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 11, no. 2, pp. 276–281, 2022. https://doi.org/10.32736/sisfokom.v11i2.1393

[14] E. Keary, “OWASP - Open Web Application Security Project,” *OWASP Found.*, no. Cc, p. 224, 2014,

[15] M. Yunus, “Analisis Kerentanan Aplikasi Berbasis Web Menggunakan Kombinasi Security Tools Project Berdasarkan Framework Owasp Versi 4,” *J. Ilm. Inform. Komput.*, vol. 24, no. 1, pp. 37–48, 2019. http://dx.doi.org/10.35760/ik.2019.v24i1.1988

[16] OWASP, “2017 Top 10,” *OWASP Found.*, p. 4, 2017.

[17] Y. Muhyidin, M. Hafid Totohendarto, E. Undamayanti, and S. Tinggi Teknologi Wastukancana, “Perbandingan Tingkat Keamanan Website Menggunakan Nmap Dan Nikto Dengan Metode Ethical Hacking Comparison of Website Security Levels Using Nmap and Nikto With Ethical Hacking Methods,” *J. Teknol.*, pp. 1–10, 2020. https://doi.org/10.51132/teknologika.v12i1.143

[18] M. Marsoni, T. U. Kalsum, and A. Kurniawan, “Analisa Implementasi Teknik Reconnaissance Pada Webserver (Studi Kasus: Upt Puskom Universitas Dehasen),” *J. Media Infotama*, vol. 12, no. 1, pp. 11–20, 2016. https://doi.org/10.37676/jmi.v12i1.268

[19] G. Guntoro, L. Costaner, and M. Musfawati, “Analisis Keamanan Web Server Open Journal System (Ojs) Menggunakan Metode Issaf Dan Owasp (Studi Kasus Ojs Universitas Lancang Kuning),” *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 5, no. 1, p. 45, 2020. https://doi.org/10.29100/jipi.v5i1.1565

[20] https://dspace.uii.ac.id/bitstream/handle/123456789/11281/13523025-Adetya Putra D-laporan skripsi.pdf?sequence=1&isAllowed=y

[21] F. Fachri, “Optimasi Keamanan Web Server Terhadap Serangan Brute-Force Menggunakan Penetration Testing,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 1, pp. 51–58, 2023. https://doi.org/10.25126/jtiik.20231015872

[22] I. G. A. S. Sanjaya, G. M. A. Sasmita, and D. M. S. Arsa, “Evaluasi Keamanan Website Lembaga X Melalui Penetration Testing Menggunakan Framework ISSAF,” *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, vol. 8, no. 2, p. 113, 2020. https://doi.org/10.24843/JIM.2020.v08.i02.p05

[23] I. Riadi, A. Yudhana, and Y. W, “Analisis Keamanan Website Open Journal System Menggunakan Metode Vulnerability Assessment,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, pp. 853–860, 2020. https://doi.org/10.25126/jtiik.2020701928

[24] G. Kusuma, “Implementasi Owasp Zap Untuk Pengujian Keamanan Sistem Informasi Akademik,” *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 16, no. 2, pp. 178–186, 2022. https://doi.org/10.47111/jti.v16i2.3995

[25] T. Syarif Revolino and D. Jatmiko Andri, “Analisis Perbandingan Metode Web Security PTES, ISSAF dan OWASP di Dinas Komunikasi Dan Informasi Kota Bandung,” *Elibrai.unikom*, p. 8, 2019.

[26] J. J. B. H. Yum Thurfah Afifa Rosaliah, “Pengujian Celah Keamanan Website Menggunakan Teknik Penetration Testing dan Metode OWASP TOP 10 pada Website SIM,” *Senamika*, vol. 2, no. September, pp. 752–761, 2021.

[27] FIRST, “Common Vulnerability Scoring System version 3.1 Specification Document Revision 1,” pp. 1–24, 2019.